Learning to Schedule Resistant to Adversarial Attacks in Diffusion Probabilistic Models Under the Threat of Lipschitz Singularities

SangHwa Hong
Department of Industrial Engineering
Seoul National University of Science and Technology
Gongreung-ro 232, Nowon-gu, Seoul, South Korea
hongsw5911@seoultech.ac.kr

Abstract

Recently, the field of generative models has advanced significantly with the introduction of Diffusion Probabilistic Models (DPMs). However, the discovery of Lipschitz Singularities within DPMs reveals a vulnerability to subtle adversarial attacks, particularly at timesteps close to zero. This paper introduces a novel approach to enhance the robustness of DPMs against adversarial attacks, specifically addressing the challenge posed by Lipschitz Singularities. By implementing a dynamic scheduling strategy of σ through Reinforcement Learning (RL), we mitigate the adverse effects stemming from adversarial attacks that exploit vulnerabilities linked to Lipschitz singularities. Experimental results demonstrate the effectiveness of our approach in maintaining high-quality image generation.

1. Introduction

Diffusion Probabilistic Models (DPMs) [17], [24], [29], serving as a foundational architecture [5], [13] for various generative models, have gained significant popularity in the computer vision community due to their ability to circumvent the complex optimization issues associated with adversarial training in Generative Adversarial Networks (GANs) [6], [16], for generative tasks. DPMs utilize a two-step approach [17] described in Figure 1: the diffusion process, which gradually transforms original data into a Standard normal distribution by adding noise, and the denoising process, which systematically removes noise to restore the original data distribution by predicting the added noise in the diffusion process.

As representative DPMs, there are the Denoising Diffusion Probabilistic Model (DDPM) and the Denoising Diffusion Implicit Model(DDIM). DDPM conducts a denoising process with whole timesteps to maintain the Markov property [17], which requires extensive GPU resources. Compared to DDPM, DDIM [29] discards the Markov property and selectively performs the denoising process at specific timesteps. While DDPM and DDIM have other distinctions, they share even more significant traits.

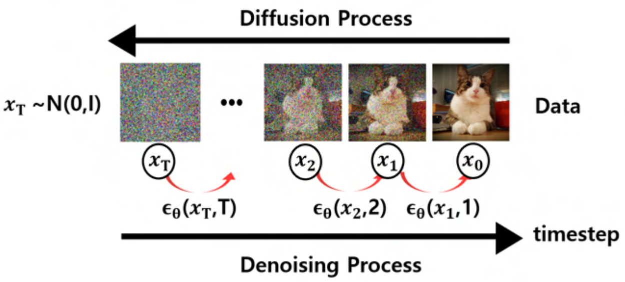

Figure 1. This figure shows the diffusion process and the denoising process where x_t $(t = T, T-1,\ \dots 2, 1)$ is defined as the vector whose elements in each dimension correspond to the pixel value in the image at timestep t, and the $\epsilon_\theta(x_t, t)$ is the parameterized network predicting the added noise in diffusion process.

In contrast to DDPM, DDIM requires the scheduling of the sigma (σ) hyperparameter as described in Figure 2 and Equation 13, which plays a crucial role as much as predicting added noise in the denoising process of the diffusion model [29]. By carefully scheduling the σ at each timestep, the model can navigate the complex trade-off between removing noise and retaining essential features of the data distribution, resulting in high-quality sample generation.

Beyond conventional strategies that implement a uniform σ scheduling for all instances during the denoising process as described in Figure 2, employing a customized σ scheduling for each instance can be more effective [31]. The assumption that all instances should have an identical schedule is flawed because it overlooks the possibility that an optimal sequence of σ values might exist for each unique instance. However, until now, the σ scheduling has been conducted without considering the individualized σ scheduling [21], [4]. Universally, applying a σ value of zero

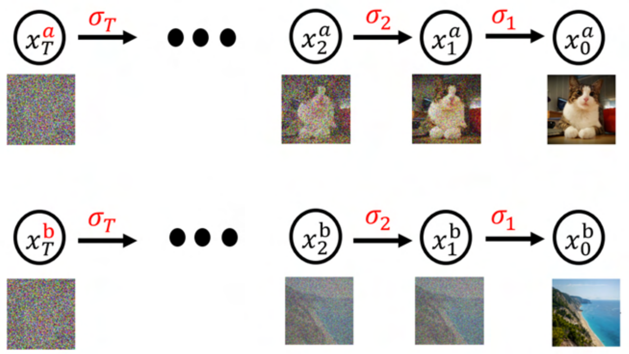

Figure 2. This figure shows that the identical σ scheduling is applied across the different instance during denoising process where the x_t^a and x_t^b $(t = T, T-1, \ldots, 2, 1)$ are defined as the vector whose elements in each dimension correspond to the pixel value in the image at timestep t, and the σ_t is the σ applied at timestep t for denoising process.

uniformly in the denoising process for every instance is acknowledged as an effective strategy. [29], [31].

Both DDPM and DDIM, recognized as central architectures for generative foundation models, have recently been spotlighted for harboring **Lipschitz Singularities** [34], which can make the foundation models built on diffusion models vulnerable to even subtle adversarial attacks [2]. During denoising processes, diffusion models transform a noised image (x_t) and a timestep (t) into a denoised image (x_{t-1}) by predicting the noise ($\epsilon_\theta(x_t, t)$) as described in Figure 1 and the Equation 13 [29]. However, concerning Equation 1 in the [34], the existence of Lipschitz Singularities makes prediction of noise ($\epsilon_\theta(x_t, t)$) unstable if even slight changes are made on input t especially when timesteps of denoising process, described in Figure 1, approach to zero. Thus, subtle adversarial attacks on timesteps near zero not only make each denoising step, outlined in Equation 13, defective but also deteriorate the subsequent denoising steps across the denoising process, consequently generating degraded x_0; refer to Figure 6.

$$\limsup_{t \to 0^+} \left| \frac{\partial \epsilon_\theta(x, t)}{\partial t} \right| \to \infty \tag{1}$$

Empirical experiments have discovered cases where the qualities of x_0 deteriorate when adversarial attacks are conducted at timesteps near zero (noise, denoted as ϵ and sampled from the normal distribution, plays a role in adversarial attacks). Due to the property of the diffusion model, the x_0 can be predicted at a certain timestep $t (1 \leq t \leq T)$, with the predicted x_0 denoted as $\hat{x}_0$; refer to Equation 10, [22]. The qualities of $\hat{x}_0$ tend to increase or remain constant as the denoising process progresses if there are no adversarial attacks on the timesteps near zero. In contrast, the qualities of $\hat{x}_0$ suddenly decrease if even slight adversarial attacks are conducted on timesteps near zero; refer to Figure 3. In addition, the quality of the x_0 usually becomes similar to degraded qualities of $\hat{x}_0$ which are predicted by x_t at timestep t near zero. Considering the adversarial attacks at timesteps near zero, Equation 1 and denoising steps detailed in Equation 13, it can be inferred that the diminished quality of the generated images is attributable to the defect of denoising steps which is caused by adversarial attacks on timesteps near zero.

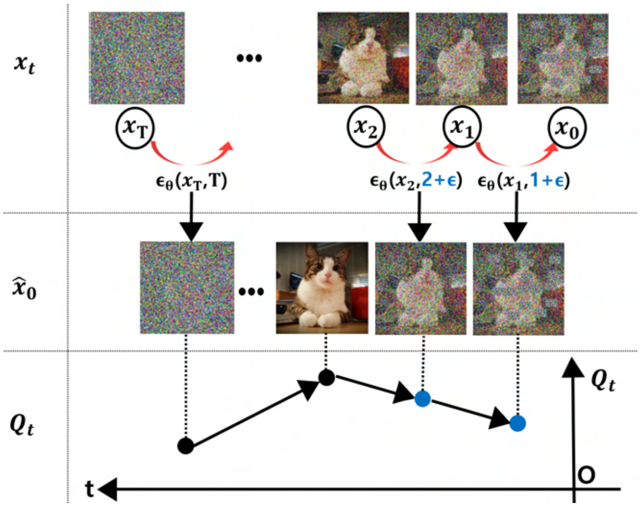
x_t
x_T
x_2
x_1
x_0
$\epsilon_\theta(x_T, T)$
$\epsilon_\theta(x_2, 2+\epsilon)$
$\epsilon_\theta(x_1, 1+\epsilon)$
$\hat{x}_0$
Q_t
Q_t
t
O

Figure 3. This figure shows that the finally denoised image x_0 is predicted by the denoised images at certain timestep $(1 \leq t \leq T)$ and qualities of predicted images are decreasing at timesteps near zero when the noise is injected into the timesteps which are inputs of network predicting noise. The $\hat{x}_0$ is the predicted images on the finally denoised image, Q_t is the quality of the predicted image and ϵ is the injected noise sampled from the normal distribution.

To develop the method for making the diffusion model robust against adversarial attacks at timesteps near zero, this paper introduces a strategy that alleviates the impact of unstable noise prediction, occurring because of the adversarial attacks on timesteps near zero during the denoising step detailed in Equation 13. It does so by implementing customized σ scheduling across each instance (x_t), in contrast to Figure 2, during denoising steps outlined in Equation 13. Scheduling σ at timesteps near zero can be effective since σ can play a role of coefficient for unstable noise prediction ($\epsilon_\theta(x_t, t)$) inside Equation 13 while stabilizing unstable noise prediction caused by adversarial attacks. To learn the optimal policy deciding the appropriate σ, Reinforcement Learning (RL) [3] is employed.

Therefore, the goal of this paper is to demonstrate that under adversarial attacks on timesteps near zero, which are associated with Lipschitz Singularities, the σ scheduled by policy can generate the better quality of x_0 than universally utilized σ scheduling approach which maintains a constant σ value of zero throughout the denoising process [29]. As shown in our experiments, the suggested approach, selectively modulating σ on each instance during denoising steps at timesteps vulnerable to adversarial attacks, can make diffusion models more robust than pre-fixed σ scheduling under adversarial attacks.

Contributions: For the first time, this paper addresses the adversarial attacks, considering the Lipschitz Singularities that can be summarized by the Equation 1. Furthermore, this paper introduces an RL-based σ scheduling strategy to enhance the robustness of diffusion models against adversarial attacks.

Rest of papers: In the related work section, the introductions of Lipschitz Singularities and the Markov Decision Process (MDP) which should be defined to apply the RL algorithm are presented. In the background section, the concepts for diffusion models are presented. In the MDP formulation section, State, Action and Reward, divided into Final Reward and Intermediate Reward, are defined. In the experiment section, it is shown by experiments that the suggested approach is effective for making the diffusion model more robust than universally utilized fixed σ scheduling under the adversarial attacks on timesteps near zero.

2. Related Work

2.1. Image Quality Evaluator: Q-ALIGN

The Q-ALIGN model introduces a novel approach to visual scoring by training Large Multi-modality Models (LMMs) to evaluate image quality, aesthetics, and video quality based on discrete, text-defined levels rather than numerical scores. This method emulates how human raters assess visual content by using categories such as ”excellent,” ”good,” ”fair,” ”poor,” and ”bad” to train the LMMs. The model achieves state-of-the-art performance across various datasets for image quality assessment (IQA), image aesthetic assessment (IAA), and video quality assessment (VQA) tasks. It demonstrates not only superior accuracy and generalization capabilities but also efficiency in training and flexibility in combining datasets from different visual assessment tasks into a unified model, the ONEALIGN.

2.2. Lipschitz Singularities in Diffusion Models

Diffusion models, pivotal in generative modeling by manipulating noise through stochastic processes, face a challenge due to ”Lipschitz singularities” near the diffusion onset, where minor changes can cause major output variations, undermining model stability and reliability. These singularities complicate model training and inference, affecting image generation quality and diversity. Addressing Lipschitz singularities is crucial for enhancing diffusion models' robustness and ensuring stable, high-quality image production.

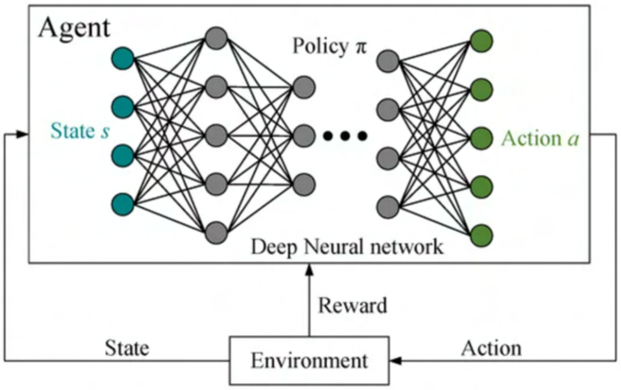

Figure 4. This figure shows the relationship among State, Action, Reward, Agent, Policy and Environment.

2.3. MDP, State, Action, Reward, Agent and RL

Utilizing the RL algorithm requires formulating the MDP [14] consisting of State, Action, and Reward [27], [32], [15], [28], [30], [18], [11]. In a Markov Decision Process (MDP), a state represents the specific conditions or status at a given time. An action refers to a choice made by the policy network of agent [1], [3] that can alter the current state. The reward is a feedback mechanism that quantifies the immediate benefit of performing a particular action from a given state, guiding the agent toward achieving its goal through a series of decisions. As described in Figure 4, these elements form the foundation of decision-making models with environments where outcomes are partly random and partly under the control of the agent. Especially, the action in this paper is defined as adjusting σ utilized for the denoising process; refer to Equation 13.

The agent receives the reward and transitioned state by interacting with the environment. The reward [10], [20], [9], [19], is the quantitative assessment on the effectiveness of action. The new state, caused by the interaction between the agent and environment, is presented to the agent for deciding the next action.

In the formed environment, the agent's objective is to explore different actions and exploit [12], [7], [25] the optimal action maximizing the accumulated reward. To optimize the policy network of the agent, the RL algorithm is utilized with the rewards collected from the interaction with the environment.

3. BACKGROUND

In this section, the fundamental concepts of diffusion models are briefly introduced. DDPM offers a latent variable model that has the following form [31]

$$p_\theta(x_0) = \int p_\theta(x_{0:T})\, dx_{1:T} \tag{2}$$

where $p_\theta(x_{0:T}) := p_\theta(x_T)\prod_{t=1}^{T} p_\theta^{(t)}(x_{t-1}|x_t)$ and $x_1, x_2, \ldots, x_T$ are latent variables that have the same dimension with original data x_0. The parameters θ are trained to make the model approximate distribution $q(x_0)$. The training is performed to maximize a variational lower bound,

$$\max_\theta \mathbb{E}_{q(x_0)}[\log p_\theta(x_0)] \leq \max_\theta \mathbb{E}_{q(x_0,x_1,\ldots,x_T)}[\log p_\theta(x_{0:T}) - \log q(x_{1:T}|x_0)] \tag{3}$$

The term $q(x_{1:T}|x_0)$ is defined as the diffusion process given by the original data x_0. In the paper [17], the diffusion process is defined as,

$$q(x_{1:T}|x_0) := \prod_{t=1}^{T} q(x_t|x_{t-1}), \tag{4}$$

where

$$q(x_t|x_{t-1}) := \mathcal{N}\left(\sqrt{\frac{\alpha_t}{\alpha_{t-1}}}x_{t-1}, \left(1-\frac{\alpha_t}{\alpha_{t-1}}\right)I\right) \tag{5}$$

The term $p_\theta(x_{0:T})$ in the Equation 2, a Markov chain from x_T to x_0, is the denoising process as it approximately matches the reverse process of the Equation 4. From the Equation 4 and Equation 5, there is a special property,

$$q(x_t|x_0) := \mathcal{N}(x_t; \sqrt{\alpha_t}x_0, (1-\alpha_t)I). \tag{6}$$

If all the conditions are considered as Gaussian with trainable parameters, the Equation 3 can be re-written as,

$$L_\gamma(\epsilon_\theta) := \sum_{t=1}^{T} \gamma_t \mathbb{E}_{x_0\sim q(x_0),\epsilon_t\sim\mathcal{N}(0,I)}\left[\left\|\epsilon_\theta\left(\sqrt{\alpha_t}x_0 + \sqrt{1-\alpha_t}\epsilon_t, t\right) - \epsilon_t\right\|\right]. \tag{7}$$

DDIM observes that the training objective of DDPM in the Equation 6 only depends on $q(x_t|x_0)$, but not directly on the joint $q(x_{1:T}|x_0)$. DDIM generalizes the DDPM as a Non-Markovian process by conditioning the x_{t-1} both x_t and x_0 instead of only x_0 as in the Equation 5. Its diffusion process can be modeled as

$$q_\sigma(x_{1:T}|x_0) = q_\sigma(x_T|x_0)\prod_{t=2}^{T} q_\sigma(x_{t-1}|x_t, x_0), \tag{8}$$

where $q_\sigma(x_{t-1}|x_t, x_0)$ is chosen to satisfy $q_\sigma(x_T|x_0) = \mathcal{N}(\sqrt{\alpha_T}x_0, (1-\alpha_T)I)$. Therefore, we have,

$$q_\sigma(x_{t-1}|x_t, x_0) = \mathcal{N}\left(\sqrt{\alpha_{t-1}}x_0 + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\left(\frac{x_t - \sqrt{\alpha_t}x_0}{\sqrt{1-\alpha_t}}\right), \sigma_t^2 I\right) \tag{9}$$

From the Equation 6, x_t can be obtained by sampling $x_0 \sim q(x_0)$ and $\epsilon_t \sim \mathcal{N}(0, I)$. By training the model $\epsilon_\theta(x_t, t)$ to predict ϵ_t at each time step, x_0 predicting function $f_\theta^{(t)}$ at timestep t is defined as,

$$f_\theta^{(t)}(x_t) = \frac{x_t - \sqrt{1-\alpha_t}\epsilon_\theta(x_t, t)}{\sqrt{\alpha_t}}. \tag{10}$$

The denoising process with prior $p_\theta(x_T) = \mathcal{N}(0, I)$ is defined as,

$$p_\theta^{(t)}(x_{t-1}|x_t) = \begin{cases} \mathcal{N}(f_\theta^{(1)}(x_1), \sigma_1^2 I) & \text{if } t = 1 \\ q_\sigma(x_{t-1}|x_t, f_\theta^{(t)}(x_t)) & \text{otherwise,} \end{cases} \tag{11}$$

The parameter θ is optimized as the Equation 12 with $q_\sigma(x_{1:T}|x_0)$ is defined in Equation 8.

$$J_\sigma(\epsilon_\theta) = \mathbb{E}_{x_{0:T}\sim q_\sigma(x_{0:T})}\Bigg[\log q_\sigma(x_T|x_0) + \sum_{t=2}^{T}\log q_\sigma(x_{t-1}|x_t, x_0) - \sum_{t=1}^{T}\log p_\theta^{(t)}(x_{t-1}|x_t) - \log p_\theta(x_T)\Bigg] \tag{12}$$

Based on the defined denoising process, given a sample x_t, we could sample the x_{t-1} as

$$x_{t-1} = \sqrt{\alpha_{t-1}}\frac{(x_t - \sqrt{1-\alpha_t}\epsilon_\theta(x_t, t))}{\alpha_t} + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\cdot\epsilon_\theta(x_t, t) + \sigma_t\epsilon_t, \tag{13}$$

where σ_t is usually configured as 0 because it is confirmed to be effective [29].

4. MDP formulation

As mentioned in the introduction, this paper suggests a strategy utilizing the RL algorithm, which firstly requires formulating MDP. In this section, The MDP is defined by the tuple (S, A, R, P, γ), where S, A, R, P and γ represent the set of states, the actions available, the reward function, the state transition and the discount rate, respectively.

Prior to presenting the MDP formulation, the notation t_c is introduced, important across the following descriptions. Referring to Figure 5, the critical timestep, t_c, marks the point where the agent begins to engage in scheduling the σ, utilized for certain denoising step described in the Equation 13. Considering that instability in noise prediction arises from adversarial attacks at timesteps near zero, scheduling σ exclusively by the agent at these timesteps is sufficient. Thus, the state, action and reward are also defined only for timesteps $t = t_c, t_{c-1}, \ldots, 2, 1$; refer to Figure 5.

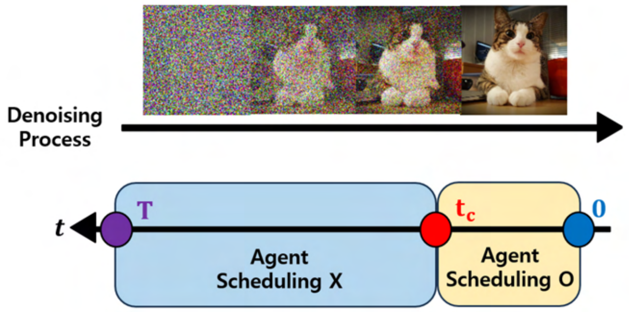

Figure 5. At timesteps in blue box ($t > t_c$), pre-defined schedule of σ is applied for denoising process. In contrast, in yellow box ($t = t_c, t_c - 1, ..., 2, 1$), the agent schedules the σ.

4.1. State

The state, denoted by s_t, is defined as the tuple consisting of the denoised image (x_t) and the timestep (t) for timesteps $t = t_c, t_{c-1}, \dots, 2, 1$; refer to the Equation 14. This definition of the state can distinguish whether the noise prediction, $\epsilon_\theta(x_t, t)$, is stable or not. Because the noise prediction is only dependent on the x_t, t, whether the noise prediction is stable or not can be also distinguished by x_t, t. With this state definition, the agent is capable of identifying effective actions for a given state.

$$s_t = (x_t, t) \tag{14}$$

4.2. Action

The action, denoted as a_t, is to decide the σ, denoted as σ_t^{Ag} and utilized in the each denoising steps specified in Equation 13 for timesteps $t = t_c, t_{c-1}, \dots, 2, 1$; refer to Equation 15. According to Equation 13, σ also can act as a coefficient of noise prediction ($\epsilon_\theta(x_t, t)$) at timesteps near zero. By selectively adjusting the σ_t^{Ag} across each instance during the denoising process, it can effectively stabilize the unstable noise prediction.

$$a_t = \sigma_t^{Ag} \tag{15}$$

4.3. Transition

The transition function P is defined as taking an input a_t given the current state s_t, and outputs the next state as the tuple of $(p_\theta^{(t)}(x_{t-1}|x_t),\ t-1)$ for timesteps $t = t_c, t_{c-1}, \dots, 2$; refer to the Equation 11. In the denoising process, as detailed by Equation 13, Since x_{t-1} is derived by $p_\theta^{(t)}(x_{t-1}|x_t)$, the next state can be also driven as the tuple of $(x_{t-1},\ t-1)$. Even though the next state is conventionally denoted as s_{t+1}, with regard to the denoising process where the timestep t decreases as the process proceeds, the next state is denoted as s_{t-1}. To sum up, the transition function taking an input a_t given the current state s_t and outputs the next state s_{t-1} is defined as the Equation 16.

$$s_{t-1} = P(a_t|s_t) = (p_\theta^{(t)}(x_{t-1}|x_t), t-1) \tag{16}$$

4.4. Reward

To define the reward that can reflect the goal presented in the introduction, components such as image quality evaluator [33] and baseline σ are required. As described in Figure 6, the image quality evaluator is the pre-trained deep learning model that scores the input image's quality where $E_{\theta_{freeze}}$ is denoted as the deep learning-based parameterized evaluator function and the Q is the quality estimated by the evaluator function. During inference time, the parameters of the model are frozen.

The baseline σ, denoted as σ_t^{Ba}, refers to the predefined sigma used for the denoising step as outlined in Equation 13 for $t = T, T-d, T-2d, \dots, t_c+d, t_c, t_c-1, t_c-2\dots, 2, 1$ where the fixed interval for denoising process is denoted as d. The universally utilized scheduling such as maintaining σ as zero is applied to σ_t^{Ba}. In contrast, The agent-determined sigma, σ_t^{Ag}, is selected by the agent based on the state s_t for timesteps $t = t_c, t_{c-1}, \dots, 2, 1$.

In the following sections, the two types of reward, Final Reward denoted as r_1 and Intermediate Reward denoted as r_t $(t > 1)$, are defined respectively by utilizing the components introduced above. By defining the reward r_t as subtraction Q_t^{Ba} from Qt^{Ag} where the Q_1^{Ag}, Q_t^{Ag}, Q_1^{Ba} and Q_t^{Ba} are denoted as qualities of x_0^a, $\hat{x}_0^a$, x_0^b and $\hat{x}_0^b$ which are described in Figure 6, we can quantitatively assess how much better the quality of x_0 is preserved under adversarial attacks on timesteps near zero when σ is scheduled with σ_t^{Ag}(agent-determined scheduling) compared to σ_t^{Ba}(fixed scheduling), for timesteps $t = t_c, t_c - 1, ..., 3, 2, 1$.

4.4.1 Final Reward

We calculate Final Reward r_1 on a_1 by the difference in the estimated qualities of the fully denoised images (x_0^a and x_0^b); refer to Figure 6 for the following. This quality comparison is conducted between the quality of the image (x_0^a) which is denoised with the σ_t^{Ag} for timesteps $t = t_c, t_c - 1, ..., 2, 1$ and σ_t^{Ba} for timesteps $t = T, T - d, ...t_c + 2d, t_c + d$ (**green arrow**), against the quality of x_0^b which is denoised with the σ_t^{Ba} for timesteps $t = T, T-d, ..., t_c+d, t_c, t_c-1, t_c-2, ...2, 1$ (**blue arrow**), given the identical x_T^b sampled by the Standard normal distribution. The r_1 is mathematically expressed by the Equation 16, 17 and 18.

$$Q_1^{Ag} = E_{\theta_{freeze}}(x_0^a|\sigma_1^{Ag}, \dots, \sigma_{t_c-2}^{Ag}, \sigma_{t_c-1}^{Ag}, \sigma_{t_c}^{Ag}, \sigma_{t_c+d}^{Ba}, \dots, \sigma_{T-d}^{Ba}, \sigma_T^{Ba}, x_T^b) \tag{16}$$

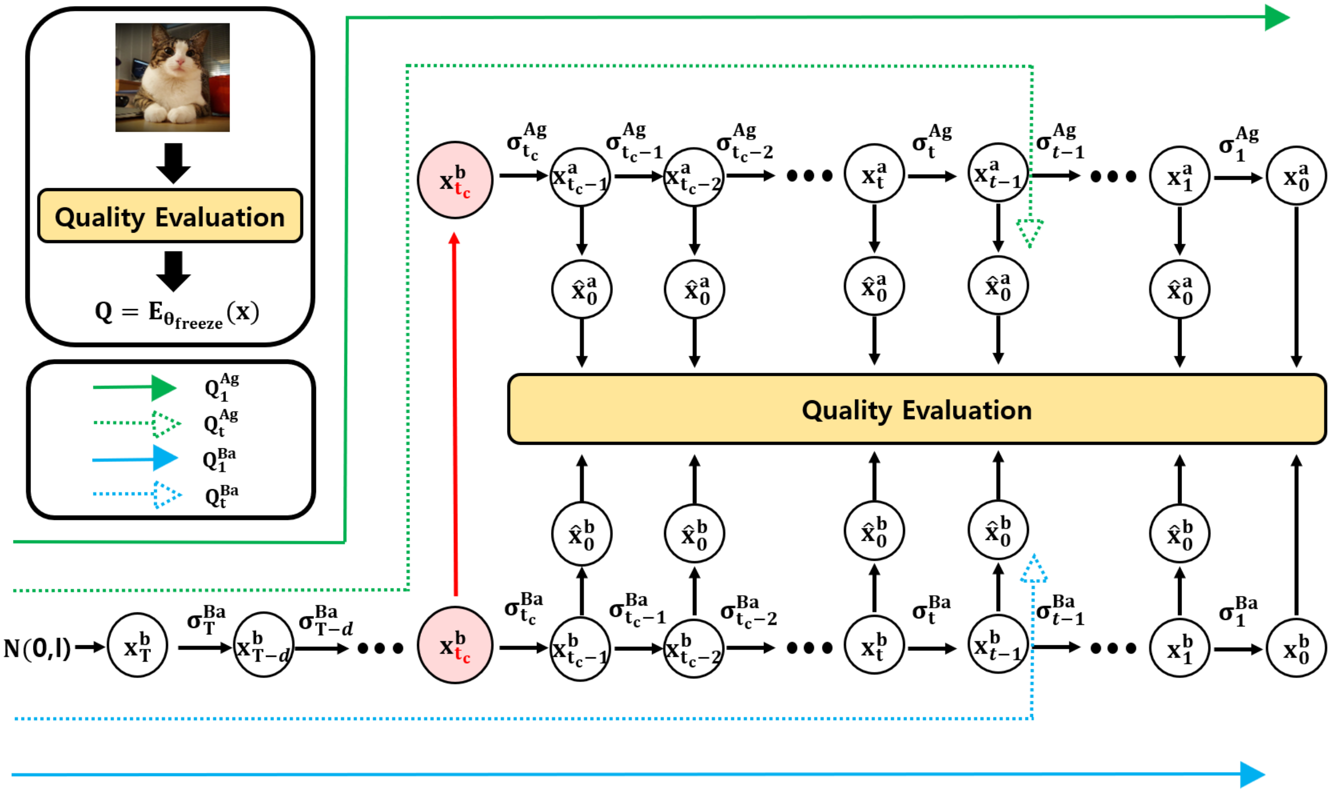

Figure 6. This figure shows how the qualities of x_0^a, $\hat{x}_0^a$, x_0^b and $\hat{x}_0^b$ are derived as the denoising process progress. x_0^a and x_0^b are images produced after full denoising steps, but scheduled by different σ scheduling.

$$Q_1^{Ba} = E_{\theta_{freeze}}(x_0^b|\sigma_1^{Ba},\dots,\sigma_{t_c-2}^{Ba}, \sigma_{t_c-1}^{Ba},\sigma_{t_c}^{Ba},\sigma_{t_c+d}^{Ba},\dots,\sigma_{T-d}^{Ba},\sigma_T^{Ba},x_T^b) \tag{17}$$

$$r_1 = Q_1^{Ag} - Q_1^{Ba} \tag{18}$$

4.4.2 Intermediate Reward

However, it is hard to quantitatively assess how much better $\sigma_t^{Ag}(t > 1)$ preserves the quality of x_0 than $\sigma_t^{Ba}(t > 1)$ because the relationship between σ at timestep $t(t > 1)$ and the quality of x_0 is complex. As the alternative, we utilize the $\hat{x}_0$ instead of x_0 where $\hat{x}_0$ is the prediction of x_0 from Equation 10 using x_{t-1} (obtained after applying a_t in the denoising step described by Equation 13; refer to Figure 6). Due to the property of the diffusion process detailed in Equation 10, $\hat{x}_0$ can be derived by utilizing the x_t whose t is a certain timestep. By evaluating the quality of $\hat{x}_0$ predicted by utilizing the x_{t-1} and t, it is possible to quantitatively assess how much better $\sigma_t^{Ag}(t > 1)$ preserves the quality of x_0 than $\sigma_t^{Ba}(t > 1)$.

Similarly, we calculate Intermediate Reward r_t on a_t, for $t = t_c, t_c - 1, ..., 3, 2$, by the difference in the estimated qualities of the images ($\hat{x}_0^a$ and $\hat{x}_0^b$) predicted from x_{t-1}^a and x_{t-1}^b, respectively; refer to Figure 6 for followings. This quality comparison is conducted between the quality of image ($\hat{x}_0^a$) predicted from x_{t-1}^a which is denoised with the σ_t^{Ag} for timesteps $t = t_c, t_c - 1, ..., t + 1, t$ and σ_t^{Ba} for timesteps $t = T, T - d, ...t_c + 2d, t_c + d$ (**green dotted arrow**), against the quality of $\hat{x}_0^b$ predicted from the x_{t-1}^b which is denoised with the σ_t^{Ba} for timesteps $t = T, T-d, ..., t_c+d, t_c, t_c-1, t_c-2, ...t+1, t$ (**blue dotted arrow**), given the identical x_T sampled by the Standard normal distribution. The r_t is mathematically expressed by the Equation 19, 20 and 21.

$$Q_t^{Ag} = E_{\theta_{freeze}}(\hat{x}_0^a|x_{t-1}^a,\sigma_t^{Ag},\dots,\sigma_{t_c-2}^{Ag}, \sigma_{t_c-1}^{Ag},\sigma_{t_c}^{Ag},\sigma_{t_c+d}^{Ba},\dots,\sigma_{T-d}^{Ba},\sigma_T^{Ba},x_T^b) \tag{19}$$

$$Q_t^{Ba} = E_{\theta_{freeze}}(\hat{x}_0^b|x_{t-1}^b,\sigma_t^{Ba},\dots,\sigma_{t_c-2}^{Ba}, \sigma_{t_c-1}^{Ba},\sigma_{t_c}^{Ba},\sigma_{t_c+d}^{Ba},\dots,\sigma_{T-d}^{Ba},\sigma_T^{Ba},x_T^b) \tag{20}$$

$$r_t = Q_t^{Ag} - Q_t^{Ba} \tag{21}$$

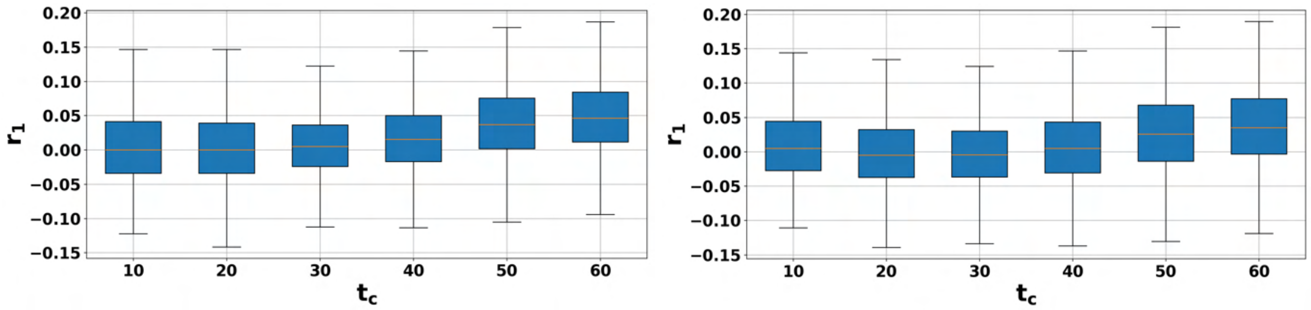

Figure 7. Left figure and Right figure correspond to the experiment result of CelebA and ImageNet, respectively. On the y-axis, the r_1 described in 4.4.1 is utilized. As the t_c where the agent starts to schedule $\sigma(\sigma_t^{Ag})$ increases, the r_1 tends to increase.

5. Experiment

The objective of the experiments is to demonstrate that in the face of adversarial attacks on timesteps near zero during the denoising process of diffusion models—a situation where noise prediction, vital for the denoising process, experiences instability, lead to negative impact on the subsequent denoising steps and ultimately result in the generation of lower-quality samples x_0—optimizing the scheduling of σ through an RL algorithm can consequently enhance the quality of the generated image x_0.

5.1. Experiment Setup

5.1.1 Software and Hardware

All experiments were conducted using Pytorch. All the experiments were conducted using a GPU server equipped with two NVIDIA RTX 3090 GPUs, 128 GB RAM, and an Intel i9-10940X CPU.

5.1.2 Dataset and Generative Model

The Generative models presented in [29], DDIM, is trained by 64x64 resolution CelebA [23] and 64x64 resolution ImageNet [8] dataset. CelebA dataset provides a diverse collection of celebrity face images while ImageNet provides a diverse set of images across a wide array of categories. This diversity challenges generative models to capture a broader spectrum of features and patterns. Therefore, this diversity helps in assessing the model's capability to generate and reconstruct complex features, making it a benchmark dataset in the field of computer vision for generative tasks. The generative model is trained using a linear scheduler, with the total number of timesteps (T) set to 1000 [29]. After training is once finished, all parameters of the generative models are frozen.

5.1.3 Training Setup for Policy Network

Adversarial attacks are executed at timesteps $t(t \leq t_c)$ during the generation of x_0 by generative models. For implementing these attacks, noise from the standard normal distribution, scaled down by a factor of 0.01 and clipped within absolute value 0.05, is injected into the timesteps. The σ_t^{Ba} for timesteps $t = t_c, t_{c-1}, \ldots, 2, 1$ are kept at zero(universally utilized σ scheduling). This scheduling of σ has been deemed effective in prior work [29] and is a common practice in research related to DDIM [31].

The setup for data collection, utilized for training the policy network and derived from interactions between the agent and the environment, is as follows. In response to adversarial attacks, the policy network produces the action a_t (σ_t^{Ag}) which is utilized in denoising steps while σ_t^{Ba} simply maintains the σ value as zero. To assess the rewards r_t for actions a_t (σ_t^{Ag}) at timesteps $t = t_c, t_c - 1, \ldots, 2, 1$, an image quality evaluator is necessary. We employ a pre-trained image quality evaluator suggested by [33], which assigns a continuous quality score to images ranging from 1 to 5.

The agent network employs a Multi-Layer Perceptron (MLP) architecture and the Proximal Policy Optimization (PPO) algorithm is employed, with an epsilon value of 0.2, as outlined in [35]. The γ is configured as 0.995.

5.1.4 Evaluation Setup for Policy Network

After training the policy network π, the agent is once finished, the parameters of the policy network is frozen for evaluation of policy. Under the adversarial attacks, sampling of images is conducted with σ scheduling from the learned policy against the fixed scheduling. Across each experiment, 500 samples are employed for statistics analysis described in Figure 7.

5.2. Result

5.2.1 Description

The r_1 quantifies how much better the quality of the generated image, x_0, is preserved from adversarial attacks when σ_t^{Ag} and σ_t^{Ba} are respectively applied during the timestep $t(t \leq t_c)$ and $t(t = T, T - d, \ldots t_c + 2d, t_c + d)$ for image generation, in comparison to the case where σ_t^{Ba} is used for

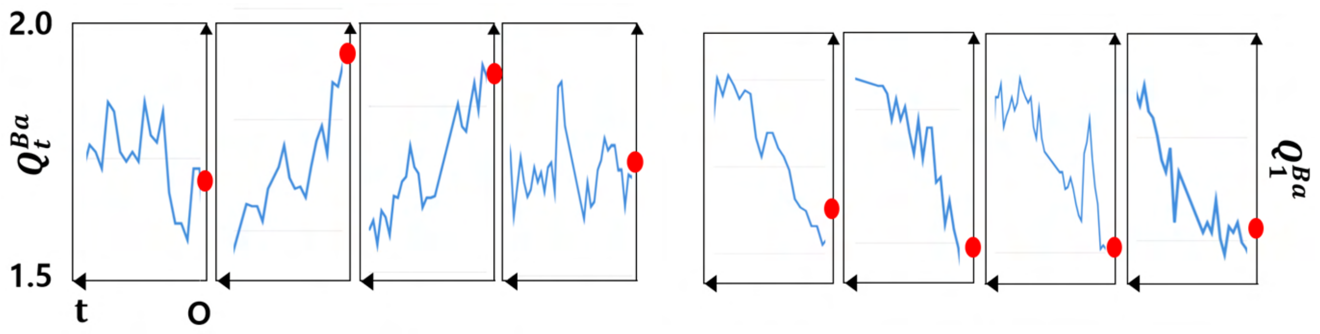

Figure 8. The first four figures from the left show the mainly observed trend for quality of images (x_0) when the Adversarial Attacks are not performed. The remaining images show the consistently observed trend of diminished image quality when adversarial attacks are performed. The y-value of the blue-colored graph corresponds to Q_t^{Ba} while the y-value of the red-colored point corresponds to Q_1^{Ba}.

all timesteps $t(1 \leq t \leq T)$. In short, it can be said that the suggested approach is effective for making the diffusion model more robust from adversarial attacks if the mean of r is greater than zero.

According to Figure 7, when the agent is trained in a situation where t_c are 10,20 and 30, there is almost no difference in preserving the quality from adversarial attacks between σ scheduling by agent and fixed-scheduler. This can be confirmed by the r_1 not distinctively deviated from the zero. In addition, this phenomenon can be confirmed from both datasets, CelebA and ImageNet.

However, when the agent is trained in situations where t_c are 40,50 and 60, there is a distinctive difference in preserving the quality from adversarial attacks between σ scheduling by the agent and fixed-scheduler. This can be confirmed by the r_1 greater than zero. This can be confirmed from both datasets CelebA and ImageNet.

In addition, there is a trend that the r_1 increases as t_c where the agent starts to schedule the σ becomes bigger. This can be confirmed from both datasets CelebA and ImageNet.

5.2.2 Analysis

The observations can be attributed to the following reasons. As t_c increases, there are more timesteps subjected to adversarial attacks, leading to a greater number of denoising steps becoming defective. Fixed scheduling of σ **fails to counteract these adversarial attacks**, resulting in a tendency for the quality of the generated x_0 to decrease as t_c increases. In contrast, because the agent learns a σ scheduling policy capable of addressing these adversarial attacks, the decline in the quality of the generated x_0 is expected to be relatively less. Because of these, the gap between the quality of the generated x_0 under the fixed scheduling of σ and the quality of the generated x_0 under the σ scheduling learned by the RL algorithm increases when adversarial attacks occur at a greater number of timesteps near zero.

6. Conclusion

This study introduces a novel strategy to enhance the robustness of DPMs against adversarial attacks, particularly addressing the challenges posed by Lipschitz Singularities. Through the strategic scheduling of the σ hyperparameter using the RL, we have demonstrated a significant improvement in the robustness and quality of image generation under adversarial attacks. Our approach not only mitigates the impact of adversarial attacks near zero timesteps but also provides a scalable solution that could be extended to other generative model architectures also facing challenge posed by Lipschitz Singularities. The empirical evidence presented underscores the potential of our method to serve as a robust framework for future research in generative modeling, paving the way for more stable and reliable generative models in the face of adversarial attacks.

7. Limitation

Although there are different noise schedulers utilized in diffusion process for training the diffusion models, this paper does not reflect the result from them such as cosine scheduler [26].

8. Appendix

Figure 8 shows the cases where quality of images, scheduled with σ maintained as zero across the whole denoising process, decreases are observed at timesteps near zero when the adversarial attacks are conducted on timesteps near zero.

References

[1] Baher Abdulhai and Lina Kattan. Reinforcement learning: Introduction to theory and potential for transport applications. *Canadian Journal of Civil Engineering*, 30(6):981–991, 2003. 3

[2] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430, 2018. 2

[3] Kai Arulkumaran, Marc Peter Deisenroth, Miles Brundage, and Anil Anthony Bharath. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017. 2, 3

[4] Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993, 2021. 1

[5] Muhammad Awais, Muzammal Naseer, Salman Khan, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, and Fahad Shahbaz Khan. Foundational models defining a new era in vision: A survey and outlook. *arXiv preprint arXiv:2307.13721*, 2023. 1

[6] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018. 1

[7] Peter Dayan and Nathaniel D Daw. Decision theory, reinforcement learning, and the brain. *Cognitive, Affective, & Behavioral Neuroscience*, 8(4):429–453, 2008. 3

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7

[9] Daniel Dewey. Reinforcement learning and the reward engineering principle. In *2014 AAAI Spring Symposium Series*, 2014. 3

[10] Jonas Eschmann. Reward function design in reinforcement learning. *Reinforcement Learning Algorithms: Analysis and Applications*, pages 25–33, 2021. 3

[11] Eugene A Feinberg and Adam Shwartz. *Handbook of Markov decision processes: methods and applications*. Springer Science & Business Media, 2012. 3

[12] Fernando Fernández and Manuela Veloso. Probabilistic policy reuse in a reinforcement learning agent. In *Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, pages 720–727, 2006. 3

[13] Matteo Ferrante, Tommaso Boccato, and Nicola Toschi. Towards neural foundation models for vision: Aligning eeg, meg and fmri representations to perform decoding, encoding and modality conversion. In *ICLR 2024 Workshop on Representational Alignment*, 2024. 1

[14] Jerzy Filar and Koos Vrieze. *Competitive Markov decision processes*. Springer Science & Business Media, 2012. 3

[15] Bob Givan and Ron Parr. An introduction to markov decision processes. *Purdue University*, 2001. 3

[16] Jie Gui, Zhenan Sun, Yonggang Wen, Dacheng Tao, and Jieping Ye. A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE transactions on knowledge and data engineering*, 35(4):3313–3332, 2021. 1

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 1, 4

[18] Qiying Hu and Wuyi Yue. *Markov decision processes with their applications*. Springer Science & Business Media, 2007. 3

[19] W Bradley Knox and Peter Stone. Combining manual feedback with subsequent mdp reward signals for reinforcement learning. In *AAMAS*, pages 5–12, 2010. 3

[20] W Bradley Knox and Peter Stone. Reinforcement learning from simultaneous human and mdp reward. In *AAMAS*, pages 475–482. Valencia, 2012. 3

[21] Max WY Lam, Jun Wang, Rongjie Huang, Dan Su, and Dong Yu. Bilateral denoising diffusion models. *arXiv preprint arXiv:2108.11514*, 2021. 1

[22] Yuseung Lee, Kunho Kim, Hyunjin Kim, and Minhyuk Sung. Syncdiffusion: Coherent montage via synchronized joint diffusions. *Advances in Neural Information Processing Systems*, 36, 2024. 2

[23] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018. 7

[24] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022. 1

[25] Pattie Maes, Maja J Mataric, Jean-Arcady Meyer, Jordan Pollack, and Stewart W Wilson. Explore/exploit strategies in autonomy. 1996. 3

[26] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pages 8162–8171. PMLR, 2021. 8

[27] Martin L Puterman. Markov decision processes. *Handbooks in operations research and management science*, 2:331–434, 1990. 3

[28] Olivier Sigaud and Olivier Buffet. *Markov decision processes in artificial intelligence*. John Wiley & Sons, 2013. 3

[29] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 1, 2, 4, 7

[30] Martijn Van Otterlo and Marco Wiering. Reinforcement learning and markov decision processes. In *Reinforcement learning: State-of-the-art*, pages 3–42. Springer, 2012. 3

[31] Yunke Wang, Xiyu Wang, Anh-Dung Dinh, Bo Du, and Charles Xu. Learning to schedule in diffusion probabilistic models. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2478–2488, 2023. 1, 2, 3, 7

[32] Douglas J White. A survey of applications of markov decision processes. *Journal of the operational research society*, 44(11):1073–1096, 1993. 3

[33] Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, et al. Q-align: Teaching lmms for visual scoring via discrete text-defined levels. *arXiv preprint arXiv:2312.17090*, 2023. 5, 7

[34] Zhantao Yang, Ruili Feng, Han Zhang, Yujun Shen, Kai Zhu, Lianghua Huang, Yifei Zhang, Yu Liu, Deli Zhao, Jingren Zhou, et al. Eliminating lipschitz singularities in diffusion models. *arXiv preprint arXiv:2306.11251*, 2023. 2

[35] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624, 2022. 7